

Klasifikasi Genre Musik Menggunakan Learning Vector Quantization dan Self Organizing Map

Luh Arida Ayu Rahning Putri^{*1}, Sri Hartati²

¹ Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana

²Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
e-mail: ^{*1}arida.arp@gmail.com, ²shartati@ugm.ac.id

Abstrak

Kemudahan dalam memperoleh file musik digital dapat menimbulkan permasalahan dalam pengelolaannya. Klasifikasi genre musik dapat membantu memberikan label genre pada file musik sehingga dapat mempermudah pengelolaan dan pencarian file musik. Permasalahan utama dalam klasifikasi genre musik adalah menemukan kombinasi fitur dan classifier yang dapat memberikan hasil terbaik dalam mengklasifikasi file musik ke dalam genre musiknya.

Penelitian ini mengklasifikasi file musik menggunakan kombinasi Learning Vector Quantization (LVQ) dan Self Organizing Map (SOM) berdasarkan pada fitur entropi koefisien wavelet. Kombinasi LVQ dan SOM terletak pada inisialisasi vektor acuan pada jaringan LVQ yang ditentukan berdasarkan hasil clustering data pelatihan menggunakan SOM. Ini diharapkan dapat mengurangi sensitivitas pemilihan vektor acuan yang dipilih langsung dari data pelatihan.

Hasil evaluasi menunjukkan bahwa klasifikasi genre musik menggunakan kombinasi LVQ dan SOM memberikan hasil yang lebih baik dibandingkan dengan menggunakan LVQ saja, namun rata-rata nilai akurasi masih rendah. Fitur entropi kurang baik dalam mengklasifikasi 10 jenis genre musik, ini ditunjukkan ketika klasifikasi dilakukan menggunakan fitur yang sama namun dengan classifier yang berbeda, akurasi yang dihasilkan juga rendah.

Kata kunci— klasifikasi genre musik, Learning Vector Quantization, Self Organizing Map, fitur entropi koefisien wavelet.

Abstract

Easiness in obtaining digital music files can cause problems in its management. Musical genre classification can help providing genre label of music files, so that management and search of music files can be simplified. The main problem in musical genre classification is to find the combination of features and classifier that can provide the best result in classifying music files into their music genre.

This research classifying music files using Learning Vector Quantization (LVQ) that combined with Self Organizing Map (SOM) based on feature of entropy of wavelet coefficients. The combination lies in the initialization of reference vectors of the LVQ which is determined based on the result of clustering the training data using SOM. This is expected could reduce the sensitivity of the reference vector selected directly from training data.

The results showed that musical genre classification using a combination of LVQ and SOM gives better results than using LVQ alone, but its accuracy is still low. Entropy features can not accurately classify the 10 genres used in this research. This was shown when classification were performed using the same feature but with different classifiers, the results were also low.

Keywords— musical genre classification, Learning Vector Quantization, Self Organizing Map, entropy feature

1. Pendahuluan

Pelabelan genre pada *file* musik merupakan hal yang penting dalam bidang *Music Information Retrieval* (MIR). Kunci utama dalam membangun MIR adalah klasifikasi, yaitu bagaimana memberikan label kepada suatu *file* musik berdasarkan genre, *mood* ataupun nama artis yang menampilkan musik tersebut [1]. Klasifikasi atau pemberian label pada suatu *file* musik ini diharapkan akan mempercepat temu kembali suatu *file* musik karena dapat mempersempit ruang pencarian hanya pada label tertentu, misalnya pada genre tertentu. Berbagai kegunaan ini merupakan alasan dibutuhkannya sistem klasifikasi genre pada *file* musik secara otomatis agar proses pelabelan *file* musik dalam jumlah besar dapat dilakukan dengan lebih efisien.

Penelitian yang paling berpengaruh dalam klasifikasi genre pada musik secara otomatis adalah [2], karena pada penelitian ini berbagai fitur berdasarkan konten untuk klasifikasi genre seperti *timbral texture*, *rhythm* dan *pitch* dikembangkan secara eksplisit [3]. Jumlah genre musik yang dilibatkan dalam penelitian ini adalah 10 genre, yaitu: Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae dan Rock. Rata-rata akurasi tertinggi yang dihasilkan [2] adalah 61% dengan menggunakan *classifier Gaussian Mixture Model* (GMM). Penelitian ini memicu timbulnya berbagai penelitian berikutnya di bidang klasifikasi genre dengan tujuan untuk menemukan fitur serta metode klasifikasi yang lebih sesuai sehingga dapat memberikan akurasi yang lebih baik.

Penelitian [3] menyatakan fitur *timbral texture* hanya menangkap statistik informasi lokal sinyal musik dari perspektif global, tetapi tidak cukup dalam mewakili informasi global musik. Hal ini disebabkan karena *timbral texture* dihitung untuk setiap *frame* dengan waktu singkat (dalam milidetik). Penelitian [3] kemudian mengusulkan sebuah teknik ekstraksi fitur baru untuk klasifikasi musik genre, yaitu *Daubechies Wavelet Coefficient Histograms* (DWCH). DWCH diperoleh dengan melakukan dekomposisi wavelet menggunakan filter Daubechies terhadap cuplikan *file* musik berdurasi 3 detik. Koefisien wavelet pada setiap *subband* hasil dari proses dekomposisi kemudian dibuat histogramnya. Setiap histogram kemudian dihitung *average*, *variance* dan *skewness*-nya. Langkah terakhir adalah menghitung nilai energi pada setiap *subband*. Klasifikasi 10 jenis genre musik berdasarkan fitur DWCH (16 fitur) yang digabungkan dengan fitur *timbral* dan MFCC (19 fitur) [2] menggunakan *classifier Support Vector*

Machine (SVM) memberikan rata-rata akurasi 78,5% [3].

Pendekatan fitur yang berbeda dilakukan oleh [4], yakni menggunakan fitur entropi dan dimensi fraktal untuk merepresentasikan sinyal musik. Kedua fitur ini didasarkan pada *information theory* yang dianggap dapat menunjukkan tingkat ketidakteraturan (*degree of irregularity*) dalam sebuah lagu yang dapat mengindikasikan genre lagu tersebut. Fitur entropi diperoleh dengan menghitung statistik entropi *frame* pada domain frekuensi (*Discrete Wavelet Transform*). Statistik dari entropi yang digunakan adalah *average entropy*, standard deviasi dari entropi, nilai maksimum entropi, nilai minimum entropi dan beda maksimum entropi dari 2 *frame* yang bersebelahan sehingga terbentuk vektor fitur berdimensi 5. Fitur ke-6 dari vektor fitur adalah dimensi fraktal dari tiap *frame* yang diperoleh dengan *box counting method*. Klasifikasi genre musik pada 3 genre musik menggunakan fitur entropi dan *classifier SVM* memberikan hasil yang cukup baik, yakni 100% untuk genre Blues dan Lounge serta 66.8% untuk genre Classical pada perbandingan 90% data pelatihan dan 10% data uji. Fitur dimensi fraktal tidak memberikan hasil yang signifikan terhadap akurasi sistem.

Learning Vector Quantization (LVQ) merupakan salah satu model Jaringan Syaraf Tiruan yang bersifat *supervised* dan dapat digunakan untuk klasifikasi. Klasifikasi genre pada musik menggunakan LVQ telah dilakukan pada [5]. Penelitian ini melakukan klasifikasi berdasarkan fitur MFCC namun dengan jumlah genre lebih sedikit dibandingkan [2]. Klasifikasi genre music dilakukan terhadap 80 *file* musik yang terbagi ke dalam 4 genre, yaitu Keroncong, Jazz, Klasik dan Rock. Fitur yang digunakan adalah MFCC dengan jumlah koefisien 13, dan menghasilkan akurasi terbaik sebesar 93.75%.

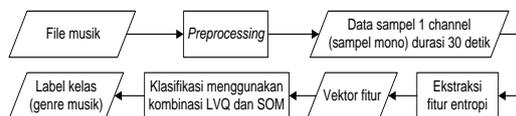
Kesederhanaan, fleksibilitas dan efisiensi merupakan beberapa alasan mengapa LVQ yang telah diperkenalkan oleh Kohonen digunakan di berbagai bidang [6]. Kelemahan dari metode ini adalah sangat bergantung atau sensitif terhadap pemilihan vektor acuan (*reference vector*) yang juga digunakan sebagai bobot awal jaringan dalam proses pelatihan LVQ. Vektor acuan merupakan perwakilan dari masing-masing kelas yang merepresentasikan suatu kelas dalam jaringan LVQ.

Penentuan vektor acuan yang paling mudah dan umum digunakan adalah dengan memilih langsung sejumlah vektor input sebagai perwakilan dari masing-masing kelas pada proses pelatihan [7]. Inisialisasi vektor acuan (bobot awal) dengan cara ini sangat sensitif terhadap tingkat akurasi karena ketidaktepatan dalam pemilihannya dapat

menghasilkan akurasi yang buruk. Alternatif lain untuk penentuan vektor acuan adalah dengan meng-*cluster* data pelatihan terlebih dahulu dengan *Self Organizing Map* (SOM), kemudian *centroid* dari masing-masing *cluster* digunakan sebagai vektor acuan dalam LVQ. Cara ini diharapkan dapat mengurangi sensitivitas pemilihan vektor acuan secara langsung dari data pelatihan [8,9]. Penelitian ini melakukan klasifikasi genre musik berdasarkan fitur entropi [4] dengan menggunakan kombinasi LVQ dan SOM sebagai *classifier*-nya.

Metode Penelitian

Data yang digunakan dalam penelitian ini adalah *data set* GTZAN yang terdiri atas 1000 potongan *file* musik dalam format AU dengan *sampling rate* 22050 Hz, mono, 16 bit dan terbagi ke dalam 10 genre musik. Masing-masing genre diwakili oleh 100 *file* musik. Proses klasifikasi genre musik dilakukan dengan langkah-langkah yang ditunjukkan oleh Gambar 1.



Gambar 1. Tahapan proses klasifikasi genre musik

2.1. Preprocessing

File musik input akan mengalami *preprocessing* terlebih dahulu. *Preprocessing* bertujuan untuk mengambil sampel dengan durasi hanya 30 detik di tengah-tengah *file* musik. Sampel *file* musik 1 *channel* dalam durasi 30 detik ini kemudian dikenakan proses ekstraksi fitur.

2.2. Ekstraksi Fitur

Proses ekstraksi fitur bertujuan untuk mengolah data sampel hasil *preprocessing* dari setiap *file* musik sehingga dihasilkan fitur entropi. Perhitungan statistik dari entropi setiap *file* musik akan membentuk vektor fitur entropi yang akan digunakan sebagai input dalam proses klasifikasi. Proses ekstraksi fitur terdiri atas proses *framing*, *Discrete Wavelet Transform* (DWT), perhitungan entropi tiap *frame* dan pembentukan vektor fitur.

2.2.1. Framing

Proses pertama yang dilakukan dalam ekstraksi fitur adalah proses *framing*. Proses *framing* bertujuan untuk memecah sampel dari *file* musik ke dalam *frame-frame* berdurasi singkat. Proses ekstraksi fitur kemudian dilakukan berdasarkan pada *frame-frame* tersebut. Proses *framing* dilakukan dengan membagi data sampel mono berdurasi 30 detik hasil *preprocessing* dari *file* musik ke dalam *frame-frame*. Proses ekstraksi fitur pada penelitian klasifikasi musik biasanya

dilakukan berbasis *frame* yang berukuran 23-46 milidetik dengan *overlap* 50% yang dianggap cukup baik jika dipertimbangkan dari sisi resolusi waktu dan redundansi [8]. Proses *framing* pada penelitian ini akan dievaluasi pada penggunaan ukuran *frame* 512 sampel (setara 23 milidetik) dan 1024 sampel (setara 46 milidetik) dengan *overlap* 50%.

2.2.2. Discrete Wavelet Transform (DWT)

Proses selanjutnya dalam ekstraksi fitur setelah proses *framing* adalah proses *Discrete Wavelet Transform* (DWT) dari masing-masing *frame*. Proses DWT terhadap masing-masing *frame* dalam penelitian ini, bertujuan untuk mentransformasi *frame-frame* yang diperoleh dari proses *framing* dari domain waktu ke domain frekuensi. Proses DWT dalam penelitian ini hanya menggunakan proses dekomposisi saja karena proses ekstraksi fitur akan dilakukan terhadap koefisien hasil dekomposisi sinyal pada *frame*.

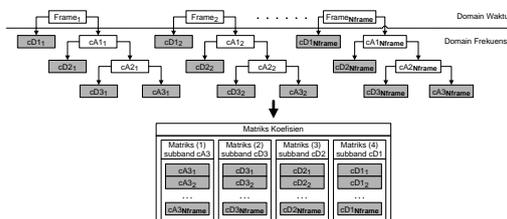
Proses ekstraksi fitur tidak hanya dilakukan pada koefisien aproksimasi yang menyatakan gambaran kasar dari sinyal asli, namun juga pada semua koefisien hasil dari proses dekomposisi baik aproksimasi maupun detail karena sinyal audio memiliki variasi frekuensi yang sangat besar yang selalu berubah setiap waktu [9]. Proses dekomposisi dilakukan terhadap sampel pada setiap *frame* dengan filter wavelet Daubechies orde 8 (Db8) seperti pada [3].

Proses DWT dari setiap *frame* akan menghasilkan sebanyak level+1 subband koefisien wavelet untuk setiap *frame*, yang terdiri atas koefisien detail pada setiap level dekomposisi dan koefisien aproksimasi pada level terakhir. Gabungan hasil DWT dari seluruh *frame* akan membentuk sebanyak level+1 matriks subband yang digabungkan lagi menjadi sebuah matriks koefisien. Ilustrasi proses DWT 3 level beserta matriks koefisien subband yang dihasilkannya ditunjukkan oleh Gambar 2.

2.2.3. Perhitungan entropi frame

Proses perhitungan nilai entropi *frame* bertujuan untuk mencari nilai entropi dari masing-masing *frame* pada *file* musik yang telah ditransformasi ke domain frekuensi dengan DWT. Perhitungan nilai entropi dari setiap *frame* mengacu pada [10], yakni diawali dengan pembentukan histogram terhadap koefisien wavelet pada masing-masing subband dari setiap *frame*, selanjutnya histogram tersebut dinormalisasi, untuk kemudian dicari nilai entropi tiap *frame* berdasarkan histogram yang telah dinormalisasi tersebut. Setiap *frame* akan menghasilkan entropi sesuai jumlah subband dalam *frame* tersebut.

Pembentukan histogram masing-masing subband dari setiap *frame* dilakukan untuk mengetahui frekuensi kemunculan suatu nilai koefisien dalam sebuah subband, yang telah dikelompokkan sebelumnya ke dalam beberapa kelompok nilai koefisien (bin). Jumlah bin pada pembentukan histogram koefisien ini dianggap sebagai jumlah derajat nilai keabuan seperti halnya pada pembentukan histogram pada citra. Koefisien wavelet tidak direpresentasikan menggunakan bit seperti pada derajat keabuan pada citra. Nilai koefisien wavelet tidak berupa nilai integer pada rentang 0 sampai dengan jumlah bit-1, melainkan berupa bilangan bulat, oleh karena itu jumlah bin yang digunakan dalam pembentukan histogram dalam penelitian ini akan dievaluasi pada beberapa nilai perpangkatan 2, yaitu 16, 32, ..., 1024.



Gambar 2. Ilustrasi proses DWT 3 level dan matriks koefisien yang dihasilkan

2.2.4. Pembentukan Vektor Fitur

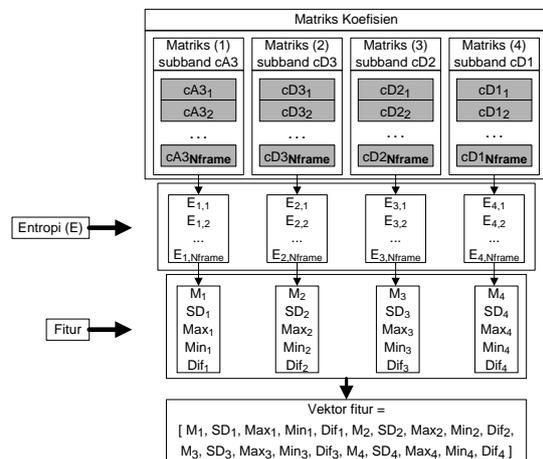
Proses terakhir dalam ekstraksi fitur adalah pembentukan vektor fitur. Pembentukan vektor fitur dilakukan dengan menghitung statistik dari entropi keseluruhan *frame* pada masing-masing subband dari sebuah *file* musik. Terdapat 5 jenis statistik yang digunakan yaitu mean (M), standard deviasi (SD), nilai maksimum (Max), nilai minimum (Min) serta beda maksimum entropi dari 2 *frame* yang bersebelahan (Dif). Masing-masing subband akan memiliki 5 buah komponen fitur, oleh karena itu dimensi vektor fitur yang dihasilkan dari masing-masing *file* musik tergantung pada jumlah subband hasil proses DWT. Ilustrasi proses pembentukan vektor fitur menggunakan DWT level 3 ditunjukkan oleh Gambar 3.

2.3 Klasifikasi Menggunakan LVQ dan SOM

Classifier yang digunakan untuk klasifikasi genre musik pada penelitian ini adalah kombinasi *Learning Vector Quantization* (LVQ) dan *Self Organizing Map* (SOM). LVQ dan SOM merupakan bagian dari Jaringan Syaraf Tiruan (JST) yang menggunakan algoritma pembelajaran kompetitif. Algoritma pembelajaran kompetitif mengkondisikan sekumpulan neuron output pada jaringan untuk berkompetisi menjadi neuron yang

aktif yang disebut dengan neuron pemenang. Proses kompetisi dari LVQ dan SOM didasarkan pada jarak Euclidean. Hanya neuron pemenang yang terpilih (aktif) saja, yakni neuron output dengan jarak Euclidean minimum terhadap data input, yang berhak memberikan respon terhadap jaringan [7].

Hasil dari pelatihan terhadap jaringan SOM dapat diperbaiki secara *supervised* menggunakan LVQ [11], sedangkan penentuan bobot awal (vektor acuan) pada LVQ dapat dilakukan dengan SOM [7]. Berdasarkan pada kedua pernyataan tersebut, maka model klasifikasi kombinasi LVQ dan SOM dalam penelitian ini dibangun dengan melakukan pelatihan jaringan menggunakan algoritma pembelajaran SOM terlebih dahulu, kemudian hasilnya dilatih lagi dengan algoritma pembelajaran LVQ.



Gambar 3. Ilustrasi proses pembentukan vektor fitur menggunakan DWT 3 level

Pelatihan jaringan pada tahap pertama menggunakan SOM akan menghasilkan bobot jaringan yang akan digunakan sebagai bobot awal untuk pelatihan jaringan menggunakan LVQ. Tahap kedua adalah pelatihan jaringan menggunakan LVQ sehingga dihasilkan bobot jaringan sebagai model klasifikasi yang akan digunakan untuk melakukan klasifikasi terhadap data uji.

2.3.1 Arsitektur jaringan kombinasi LVQ dan SOM

Aristektur kombinasi jaringan LVQ dan SOM ditunjukkan oleh Gambar 4. Jaringan SOM dan LVQ memiliki jumlah neuron input dan output yang sama. Jumlah neuron input (*n*) sama dengan dimensi vektor fitur, sedangkan jumlah neuron output (*m*) tergantung pada jumlah *codebook* per

kelas yang digunakan, yakni jumlah kelas dikalikan dengan jumlah *codebook* per kelas. Setiap neuron input (I_u , $u=1,2,\dots,n$) terhubung dengan masing-masing neuron output (O_v , $v=1,2,\dots,m$) oleh sebuah bobot $w_{u,v}$. Jaringan LVQ dan SOM akan memiliki m (sesuai jumlah neuron output) buah bobot, dengan masing-masing bobot direpresentasikan dengan vektor berdimensi n (sesuai jumlah neuron input).

Perbedaan dari jaringan LVQ dan SOM hanya terletak pada lapisan output jaringan SOM yang memiliki struktur topologi map output, dalam penelitian ini topologi map yang digunakan berbentuk segi empat. Nilai bobot jaringan hasil pelatihan jaringan SOM akan digunakan sebagai bobot awal untuk jaringan LVQ, jumlah serta label kelas dari neuron output pada jaringan SOM akan digunakan juga untuk pelatihan jaringan LVQ.

2.3.2. Pelatihan jaringan kombinasi LVQ dan SOM

Pelatihan jaringan kombinasi LVQ dan SOM diawali oleh pelatihan jaringan SOM yang mengacu pada [11]. Proses pelatihan SOM diawali dengan inisialisasi parameter-parameter jaringan yaitu inisialisasi koordinat spasial tiap neuron pada *map* output, inisialisasi bobot awal jaringan, inisialisasi nilai awal radius ketetanggaan (σ_0), inisialisasi fungsi penurunan nilai radius ketetanggaan, inisialisasi nilai awal *learning rate* (α_0), inisialisasi fungsi penurunan nilai *learning rate* serta toleransi kondisi penghentian iterasi (δ).

Proses pelatihan SOM dilakukan secara iteratif. Langkah-langkah yang dilakukan dalam setiap iterasi ke- t adalah:

1. Hitung nilai *learning rate* (αt) dan radius ketetanggaan (σt) berdasarkan nilai perubahan nilai *learning rate* dan nilai perubahan nilai radius ketetanggaan pada sesuai fungsi penurunan yang sudah dinisialisasi.
2. Pilih sebuah data input dari data pelatihan, kemudian hitung jarak Euclidean dari vektor data input terhadap vektor bobot dari masing-masing neuron output.
3. Tentukan neuron pemenang berdasarkan nilai minimum dari jarak Euclidean yang telah dihitung pada langkah 2.
4. Untuk setiap neuron output v tentukan nilai hubungan ketetanggaan antara neuron tersebut dengan neuron pemenang menggunakan fungsi Gaussian (h) seperti pada persamaan (1).

$$h_{v,c}(t) = \exp\left(-\frac{d_{v,c}^2}{2\sigma^2(t)}\right), \quad t = 0,1,2, \dots \quad (1)$$

dimana adalah $d_{v,c}$ jarak spasial antara neuron v dengan neuron pemenang c yang dihitung dengan persamaan

dimana adalah $d_{v,c}$ jarak spasial antara neuron v dengan neuron pemenang c yang dihitung dengan persamaan (2). Koordinat dari neuron output v dan neuron pemenang dinotasikan dengan r_v dan r_c .

$$d_{v,c}^2 = \|r_v - r_c\|^2 \quad (2)$$

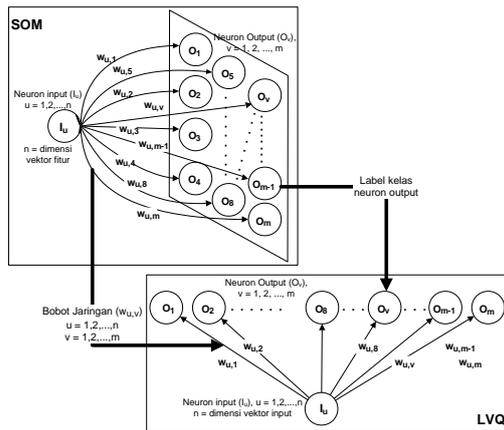
5. Untuk setiap neuron output v lakukan perbaikan nilai vektor bobot (w_v) berdasarkan persamaan (3).

$$w_v(t+1) = w_v(t) + \alpha(t)h_{v,c}(t)[I(t) - w_v(t)] \quad (3)$$

6. Ulangi langkah 1 sampai dengan 5 sampai kondisi penghentian proses pembelajaran tercapai.

Kondisi penghentian iterasi pembelajaran dilihat dari perubahan nilai vektor bobot dan nilai *learning rate*. Jika perubahan nilai vektor bobot dari masing-masing neuron hasil iterasi saat ini dibandingkan dengan nilai pada iterasi sebelumnya kurang dari toleransi perubahan δ atau jika *learning rate* (αt) nilainya kurang dari δ maka proses pembelajaran dihentikan. Saat kondisi ini tercapai bobot jaringan dianggap sudah stabil karena tidak mengalami perubahan yang signifikan.

Pelatihan SOM bersifat *unsupervised*, sehingga dalam melakukan pelatihan tidak diperlukan adanya kelas target, namun pelatihan LVQ bersifat *supervised*, oleh karena itu neuron output hasil pelatihan SOM perlu diberi label kelas untuk dapat dilatih menggunakan algoritma pelatihan LVQ. Pemberian label kelas pada *cluster* (neuron output) hasil pelatihan SOM dilakukan dengan *majority voting*. Data pelatihan dimasukkan ke dalam salah satu *cluster* berdasarkan jarak Euclidean, kemudian label kelas ditentukan dari label kelas terbanyak yang dimiliki oleh data pelatihan yang menjadi anggota suatu *cluster*. Bobot jaringan dan label kelas dari neuron output ini kemudian digunakan dalam jaringan LVQ. Pelatihan kemudian dilanjutkan menggunakan jaringan dan algoritma pelatihan LVQ.



Gambar 4. Arsitektur jaringan kombinasi LVQ dan SOM.

Pelatihan LVQ dalam kombinasi LVQ dan SOM tidak membutuhkan inisialisasi vektor acuan (*reference vector*), yaitu nilai bobot awal dan label dari bobot, karena bobot jaringan dan label kelas untuk masing-masing vektor bobot (neuron output) menggunakan hasil dari pelatihan SOM. Nilai awal *learning rate* (α_0) dan fungsi penurunan *learning rate* serta toleransi kondisi penghentian iterasi (δ) sama dengan parameter yang digunakan pada pelatihan SOM.

Proses pembelajaran LVQ juga dilakukan secara iteratif menggunakan langkah-langkah sebagai berikut:

1. Hitung besarnya nilai *learning rate* untuk iterasi ke- t (α_t) berdasarkan fungsi penurunan *learning rate*.
2. Pilih sebuah data input dari data pelatihan, kemudian hitung jarak Euclidean dari vektor data input terhadap vektor bobot dari masing-masing neuron output.
3. Tentukan neuron pemenang berdasarkan nilai minimum dari jarak Euclidean yang telah dihitung pada langkah 2.
4. Lakukan perbaikan nilai vektor bobot dari neuron pemenang berdasarkan persamaan (4) dan persamaan (5).
 Jika vektor input I dan neuron pemenang w_c memiliki kelas yang sama, maka:

$$w_c(t + 1) = w_c(t) + \alpha(t)[I(t) - w_c(t)]$$
 Jika vektor input I dan neuron pemenang w_c memiliki kelas yang berbeda, maka:

$$w_c(t + 1) = w_c(t) - \alpha(t)[I(t) - w_c(t)]$$
5. Ulangi langkah 1 sampai dengan 4 sampai kondisi penghentian proses pembelajaran tercapai.

Kondisi penghentian iterasi dalam proses pembelajaran LVQ sama dengan kondisi penghentian pada SOM. Hasil dari pelatihan

kombinasi LVQ dan SOM adalah suatu model klasifikasi berupa bobot dari jaringan dan label kelas dari masing-masing bobot (neuron output) yang akan digunakan untuk melakukan klasifikasi pada data baru di luar data pelatihan.

3. Hasi dan Pembahasan

Proses evaluasi pada penelitian ini dilakukan untuk memperoleh fitur dan model klasifikasi (*classifier*) terbaik dalam melakukan klasifikasi genre musik yang diperoleh dengan memilih nilai parameter-parameter terbaik untuk proses ekstraksi fitur dan pembangunan *classifier*. Kinerja *classifier* juga akan dievaluasi dengan membandingkan kinerja *classifier* pada penelitian ini dengan *classifier* lainnya yang sudah umum digunakan. Proses evaluasi dilakukan dengan *k-fold cross validation* dan metrik yang digunakan adalah rata-rata akurasi yang dihasilkan dari proses *cross validation*. *Classifier* dengan akurasi terbaik kemudian digunakan untuk melakukan klasifikasi genre musik terhadap data uji.

3.1 Pemilihan Parameter untuk Proses Ekstraksi Fitur

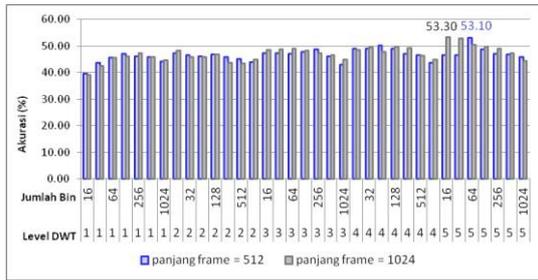
Parameter yang dievaluasi untuk proses ekstraksi fitur adalah ukuran *frame*, level DWT dan jumlah bin histogram. Hasil evaluasi untuk parameter ekstraksi fitur ditunjukkan oleh Gambar 5. Ukuran *frame* dievaluasi dengan nilai parameter 512 dan 1024 menggunakan DWT level 1 sampai dengan 5 dan jumlah bin histogram dari 16 sampai dengan 1024. Akurasi terbaik diperoleh menggunakan ukuran *frame* 1024, level DWT 5 dan jumlah bin histogram 16. Hasil evaluasi parameter proses ekstraksi ini akan digunakan sebagai acuan untuk memilih fitur yang digunakan untuk evaluasi selanjutnya, yaitu untuk pemilihan parameter terbaik dalam pembangunan *classifier*.

3.2 Pemilihan Parameter untuk Pembangunan Classifier

Parameter yang dievaluasi untuk pembangunan *classifier* adalah jumlah *codebook* per kelas dan *learning rate*. Kombinasi LVQ dan SOM akan dicoba menggunakan 3 cara yang berbeda, perbedaannya terletak pada pelatihan jaringan SOM. Pemilihan parameter dilakukan pada ketiga jenis kombinasi LVQ dan SOM.

Kombinasi LVQ dan SOM yang pertama (LVQ-SOM₁) dilakukan dengan menentukan nilai bobot awal secara acak (*random*) pada rentang [0,1], kemudian jaringan SOM dilatih dengan seluruh data pelatihan. Pelatihan dilakukan sampai kondisi berhenti terpenuhi, setelah itu dilakukan pelabelan masing-masing neuron output dengan genre musik berdasarkan *majority voting*, setelah itu jaringan dilatih lagi menggunakan algoritma

pembelajaran LVQ menggunakan seluruh data pelatihan.



Gambar 5. Hasil evaluasi parameter ekstraksi fitur

Kombinasi LVQ dan SOM yang kedua (LVQ-SOM₂) dilakukan dengan melatih jaringan SOM per kelas. Pelatihan per kelas menggunakan map output 1 dimensi dengan jumlah neuron output sejumlah *codebook* per kelas yang diinginkan. Pelatihan SOM dilakukan sebanyak jumlah kelas dengan setiap kali pelatihan menggunakan data pelatihan dari kelas yang bersesuaian saja. Setiap neuron output (vektor bobot) diberi label sesuai kelas yang akan dilatih. Bobot awal untuk pelatihan setiap kelas tetap ditentukan secara acak dalam rentang [0,1]. Di akhir pelatihan SOM, bobot jaringan dan juga label bobot hasil pelatihan pada masing-masing kelas digabungkan membentuk lapisan output dengan jumlah neuron output (vektor bobot) dan label bobot sebanyak jumlah *codebook* per kelas \times jumlah kelas. Jaringan ini kemudian dilatih dengan algoritma pembelajaran LVQ menggunakan seluruh data pelatihan.

Kombinasi LVQ dan SOM yang ketiga (LVQ-SOM₃) mirip dengan kombinasi yang kedua, perbedaannya terletak pada pemilihan nilai bobot awal jaringan SOM. Pelatihan tetap dilakukan per kelas, namun bobot awal jaringan SOM dipilih secara acak dari data pelatihan dalam kelas tersebut, label dari neuron output juga akan menyesuaikan dengan label kelas yang dilatih.

Jumlah *codebook* per kelas dievaluasi pada nilai 5, 7 dan 10. *Codebook* menyatakan perwakilan masing-masing kelas dalam neuron output. Jumlah *codebook* mempengaruhi jumlah neuron output dalam jaringan LVQ, yang juga merupakan jumlah neuron output pada jaringan SOM. Jumlah neuron output dalam jaringan output adalah sama dengan jumlah *codebook* tiap kelas dikalikan dengan banyaknya kelas. Hasil evaluasi untuk parameter jumlah *codebook* per kelas ditunjukkan pada Tabel 1.

Tabel 1. Hasil evaluasi untuk parameter jumlah *codebook* per kelas

Jumlah <i>Codebook</i>	Akurasi <i>Classifier</i> (%)			
	LVQ	LVQSO M ₁	LVQSO M ₂	LVQSO M ₃
5	50.10	50.33	51.90	53.03
7	52.87	50.30	51.23	51.03
10	51.67	52.47	53.37	52.43

Tabel 1 menunjukkan, meningkatnya jumlah *codebook* tidak selalu meningkatkan nilai akurasi. Rata-rata akurasi terbaik pada *classifier* LVQ diperoleh dengan menggunakan 7 *codebook*, pada *classifier* LVQ-SOM₁ dan LVQ-SOM₂ diperoleh dengan menggunakan 10 *codebook*, sedangkan pada *classifier* LVQ-SOM₃ akurasi terbaik diperoleh dengan menggunakan 5 *codebook* pada masing-masing kelas.

Proses evaluasi kemudian dilanjutkan untuk memilih nilai parameter *learning rate* terbaik. Pemilihan nilai *learning rate* terbaik dilakukan terhadap keempat *classifier* menggunakan jumlah *codebook* sesuai jumlah *codebook* terbaik dari masing-masing *classifier* yang diperoleh melalui proses evaluasi sebelumnya. Proses evaluasi dimulai dengan penggunaan nilai *learning rate* 0.1 kemudian diturunkan ataupun dinaikkan secara perlahan. Hasil evaluasi parameter *learning rate* ditunjukkan oleh Tabel 2.

Akurasi tertinggi yang dihasilkan oleh masing-masing *classifier* diperoleh pada nilai *learning rate* yang berbeda. Akurasi tertinggi dari *classifier* LVQ diperoleh pada saat *learning rate* bernilai 0.1, yakni 52.87%. Akurasi tertinggi dari *classifier* LVQ-SOM₁ diperoleh pada saat *learning rate* bernilai 0.35, yakni 54.2%. Akurasi tertinggi dari *classifier* LVQ-SOM₂ diperoleh pada saat *learning rate* bernilai 0.4, yakni 53.93%. Akurasi tertinggi dari *classifier* LVQ-SOM₃ diperoleh pada saat *learning rate* bernilai 0.15, yakni 54.23%.

Akurasi tertinggi di antara keempat *classifier* dihasilkan oleh *classifier* LVQ-SOM₃. Hasil evaluasi secara umum menunjukkan bahwa akurasi tertinggi yang dihasilkan oleh *classifier* kombinasi LVQ dan SOM memberikan akurasi yang lebih baik dibandingkan *classifier* LVQ.

3.3 Perbandingan Kinerja Antar Classifier

Perbandingan kinerja dilakukan antara *classifier* LVQ serta kombinasi LVQ dan SOM, dengan *classifier* lain yang umum digunakan dalam permasalahan klasifikasi, yaitu *k-Nearest Neighbour* (kNN) serta *Multi Layer Perceptron Backpropagation* (MLP-BP). Model klasifikasi

dari *classifier* kNN dan MLP-Backpropagation yang digunakan dalam proses evaluasi ini dibangun dengan memanfaatkan aplikasi WEKA. *Classifier* kNN menggunakan nilai k=1 (NN), k=5, k=7 dan k=10, sedangkan *classifier* MLP-BP menggunakan nilai parameter yang sudah dipilih oleh aplikasi WEKA.

Perbandingan kinerja pada beberapa *classifier* ditunjukkan pada Tabel 3. Semua *classifier* menghasilkan rata-rata akurasi yang masih rendah dalam mengklasifikasikan 10 genre musik dalam *data set* GTZAN. Hal ini mengindikasikan fitur entropi yang digunakan dalam penelitian ini kurang baik dalam membedakan kesepuluh jenis genre. Tabel 3 juga menunjukkan bahwa *classifier* LVQ maupun kombinasi LVQ dan SOM memberikan hasil yang sedikit lebih rendah dibandingkan dengan *classifier* MLP-Backpropagation, namun lebih baik daripada *classifier* kNN.

Tabel 2. Hasil evaluasi untuk parameter *learning rate*

Akurasi (%) Classifier	Learning Rate									
	0	0	0	0	0	0	0	0	0	0
8	9	1	5	2	2	3	3	4	4	5
LVQ	5	5	5	5	5	5	5	4	4	4
	1	1	2	2	2	1	1	1	7	7

	4	4	8	3	5	8	4	1	7	1
	0	0	7	0	0	3	7	7	7	0
LVQ	5	5	5	5	5	5	5	5	4	4
SO	0	0	2	1	2	3	2	4	1	9
M ₁
	6	0	4	5	8	9	5	2	3	7
	7	0	7	0	7	0	7	0	7	3
LVQ	5	5	5	5	5	5	5	5	5	4
SO	2	2	3	2	1	2	3	3	3	1
M ₂
	5	4	3	4	4	1	2	3	9	6
	7	7	7	7	7	0	0	0	3	7
LVQ	5	5	5	5	5	5	5	4	4	4
SO	2	2	3	4	3	3	2	2	8	7
M ₃
	0	6	0	2	1	4	3	3	4	4
	0	3	3	3	0	3	3	3	3	0

Tabel 3. Perbandingan kinerja beberapa *classifier*

Akurasi Classifier (%)								
L	LVQ	LVQ	LVQ	1-	5-	7-	1	
V	SO	SO	SO	N	N	N	0-	ML
Q	M ₁	M ₂	M ₃	N	N	N	N	P-BP
5				4	5	5	5	
2.				9.	3.	2.	2.	
8	54.2	53.9	54.2	1	7	8	2	55.
7	0	3	3	0	0	0	0	90

3.4 Klasifikasi Data Uji

Classifier dengan akurasi terbaik, yakni LVQ-SOM₃ dievaluasi untuk melakukan klasifikasi genre musik terhadap data uji. Data uji yang digunakan adalah *file* musik di luar *data set* pelatihan, yakni 30 *file* musik untuk masing-masing genre dari 10 genre yang digunakan dalam penelitian.

Klasifikasi terhadap data uji dilakukan dengan pengambilan sampel sepanjang 30 detik untuk ekstraksi fitur pada 5 posisi yang berbeda. Posisi 1, ekstraksi fitur dilakukan pada sampel yang diambil di awal *file* musik. Posisi 2, ekstraksi fitur dilakukan pada sampel yang diambil di tengah-tengah *file* musik. Posisi 3, ekstraksi fitur dilakukan pada sampel yang diambil di akhir *file* musik. Posisi 4, ekstraksi fitur dilakukan pada sampel yang diambil di awal, di tengah dan di akhir *file* musik, genre musik kemudian ditentukan berdasarkan nilai modus dari hasil klasifikasi pada ketiga posisi tersebut. Posisi 5, ekstraksi fitur dilakukan pada sampel yang diambil setiap 30 detik di sepanjang *file* musik, kemudian genre musik ditentukan berdasarkan nilai modus dari hasil klasifikasi menggunakan fitur yang diekstraksi dari setiap potongan sampel di sepanjang *file* musik tersebut.

Hasil klasifikasi terhadap data uji ditunjukkan oleh Tabel 4. Hasil klasifikasi terbaik diperoleh dengan menggunakan posisi kedua, yaitu klasifikasi dilakukan berdasarkan fitur yang diekstraksi dari potongan sampel musik sepanjang 30 detik di tengah-tengah *file* musik.

Tabel 4. Hasil klasifikasi terhadap data uji

Classifier	Posisi pengambilan sampel				
	A	Tengah	Akhir	Awal, Tengah, Akhir	Sepanjang file
LVQ-SOM ₃	wa	l	gah	hir	ng file
Akurasi (%)	36.67	45.67	38.67	42	44.33

4. Kesimpulan

Berdasarkan hasil penelitian maka dapat disimpulkan bahwa klasifikasi genre musik menggunakan kombinasi *Learning Vector Quantization* (LVQ) dan *Self Organizing Map* (SOM) terhadap 10 genre musik memberikan hasil yang lebih baik dibandingkan dengan menggunakan hanya LVQ saja. Klasifikasi genre musik dengan kombinasi LVQ dan SOM paling baik menggunakan pelatihan jaringan SOM per kelas serta bobot awal jaringan SOM dipilih secara acak dari data pelatihan pada kelas yang bersesuaian.

Akurasi yang dihasilkan pada klasifikasi 10 genre musik berdasarkan fitur entropi menggunakan kombinasi LVQ dan SOM masih rendah. Hal ini kemungkinan disebabkan oleh ketidaksesuaian fitur yang digunakan, karena ketika klasifikasi genre musik dilakukan dengan *classifier* lain (kNN dan MLP *Backpropagation*) akurasi yang dihasilkan tetap rendah.

5. Saran

Perlu dilakukan penelitian lebih lanjut untuk mencari fitur yang lebih sesuai dalam melakukan klasifikasi terhadap 10 genre musik pada *data set* GTZAN, namun tetap menggunakan kombinasi LVQ dan SOM sebagai *classifier*-nya.

DAFTAR PUSTAKA

- [1] Fu, Z., Lu, G., Ting, K.M dan Zhang, D., 2011, A Survey of Audio-Based Music Classification and Annotation, *IEEE Transactions on Multimedia*, No 2, Vol 13, hal 303-319.
- [2] Tzanetakis, G. dan Cook, P., 2002., Musical Genre Classification of Audio Signals, *IEEE Transactions on Speech and Audio Processing*, No 5, Vol 10, hal 293-302.
- [3] Li, T., Ogihara, M. dan Li, Q., 2003, A Comparative Study on Content-Based Music Genre Classification, *Proc. 26th Annu. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Toronto.
- [4] Goulart, A., Guido, R. dan Maciel, C., 2012, Exploring Different Approaches for Music Genre Classification, *Egyptian Informatic Journal*, Vol 13, hal 59-63.
- [5] Fansuri, M.R., 2011, Klasifikasi Genre Musik Menggunakan Learning Vector Quantization (LVQ), *Tesis*, FMIPA, Institut Pertanian Bogor, Bogor.
- [6] Ghosh, A., Biehl, M. dan Hammer, B., 2006, Performance Analysis of LVQ Algorithm: A Statistical Physics Approach, *Journal Neural Networks – 2006 Special Issue*, Vol 19, hal 817-829.
- [7] Fausett L. 1994. *Fundamental of Neural Network Architectures, Algorithm, and Applications*. Prentice Hall, New Jersey.
- [8] Weihs, C., Ligges, U., Morchen, F., dan Mullensiefen, D., 2007, Classification in Music Research, *Advance in Data Analysis and Classification*, Vol 1, hal 255–291.
- [9] Li, G. dan Khokhar, A.A., 2000, Content-based indexing and retrieval of audio data using wavelets, *IEEE Int. Conference on Multimedia and Expo (ICME)*, New York, 30 Juli-2 Agustus.
- [10] Gonzalez, R.C. dan Woods, R.E., 2002, *Digital Image Processing, 2nd Ed.*, Prentice Hall, New Jersey.
- [11] Haykin, S., 2005, *Neural Networks: A Comprehensive Foundation Second Edition*, Prentice Hall, Delhi.